Week 6 Problem Set:

For each of the following problems give:

A. **The function that represents the quantity to be optimized**. This includes specifying the parameters. E.g. F(I, J): this represents the maximum number of coins that the robot can pick up from a rectangle of size IxJ made up of the first I rows and first J columns..
B. **The recurrence formula**: the formula that specifies the recurrence relation that represents the relationship between larger problems and smaller problems. E.g. for the robot problem
F(I, J):= max { F(I, J) , F(I, J)} + δ i,j where δ i,j = 1 if there is a coin in the i,j square and 0 if not
C. **The base cases**: E.g. F(0,J) = F(I, 0) = 0 for all I from 0 to N and all J from 0 to M
D. **The final solution:** E.g. F(N,M) if the original problem is stated as an NxM board
E. **The pseudo code** for the iterative algorithm to fill in a table that would be used for a bottom up dynamic programming solution

1. Given an unlimited supply of coins of denominations $d_1 < d_2 < : : : < d_n$, we wish to make change for a value v; that is, we wish to find a set of coins whose total value is v. **This might not be possible: for instance, if the denominations are 5 and 10 then we can make change for 15 but not for 12**. Give a dynamic programming algorithm for the following problem: You are given $d_1 < d_2 < : : : < d_n$ and v. You must answer the following question:
Is it possible to make change for v using coins of denominations $d_1 < d_2 < : : : < d_n$?
2. Firestones is considering opening a series of restaurants along Highway 1. The n possible locations are along the highway, and the distances of these locations from the downtown San Luis Obispo are, in miles and in increasing order, $m_1, m_2, \ldots, m_n$. The constraints are:
   - At each location, Firestones may open at most one restaurant. The expected profit from opening a restaurant at location i is p_i, where $p_i > 0$ and i = 1; 2; : : : ; n.
   - Any two restaurants must be at least k miles apart, where k is a positive integer.

Give a dynamic programming algorithm to compute the maximum expected total profit subject to the given constraints.

3. You are going on a long trip. You start on the road at mile post 0. Along the way there are n hotels, at mile posts $a_1 < a_2 < \ldots < a_n$, where each a_i is measured from the starting point. The only places you can stop are at these hotels, but you can choose which of the hotels you stop at. You must stop at the final hotel (at distance a_n), which is your destination.
   You would ideally like to travel 300 miles a day, but this may not be possible (depending on the spacing of the hotels). If you travel x miles during a day, the penalty for that day is $(300 - x)^2$. You want to plan your trip to minimize the total penalty-that is, the sum, over all travel days, of the daily penalties.

Give an efficient algorithm that determines the optimal sequence of hotels at which to stop.

4. Pebbling a checkerboard. Given a checkerboard which has 4 rows and n columns, and has an integer written in each square. We are also given a set of 2n pebbles, and we want to place some or all of these on the checkerboard (each pebble can be placed on exactly one square) so as to maximize the sum of the integers in the squares that are covered by pebbles. There is one constraint: for a placement of pebbles to be legal, no two of them can be on horizontally or vertically adjacent squares (diagonal adjacency is fine).

(a) Determine the number of legal patterns that can occur in any column (in isolation, ignoring the pebbles in adjacent columns) and describe these patterns. Call two patterns compatible if they can be placed on adjacent columns to form a legal placement. Let us consider subproblems consisting of the first k columns $1 \leq k \leq n$. Each subproblem can be assigned a type, which is the pattern occurring in the last column.
(b) Using the notions of compatibility and type, give an O(n)-time dynamic programming algorithm for computing an optimal placement.